Brain Computer Interaction

Feature Translation – Decision Trees

<u>Course Instructors</u>

Dr. Annushree Bablani

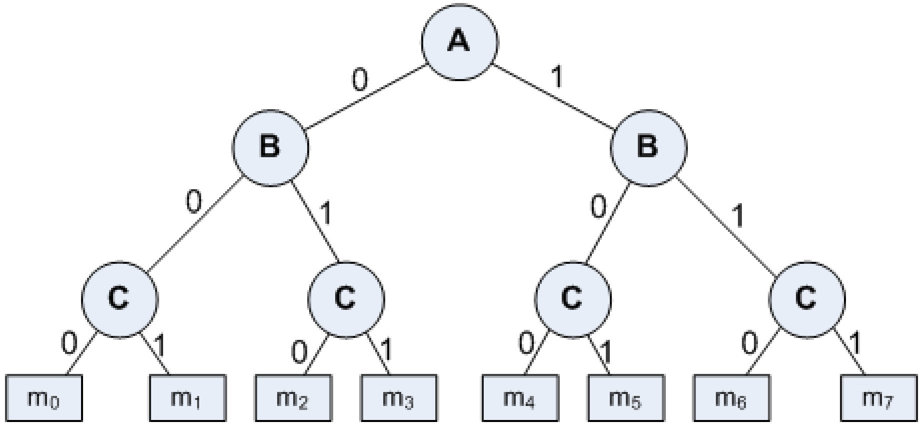
Acknowledgements: Dr. Sreeja S R

Basic Concept

- A Decision Tree is an important data structure known to solve many computational problems

Example 8.1: Binary Decision Tree

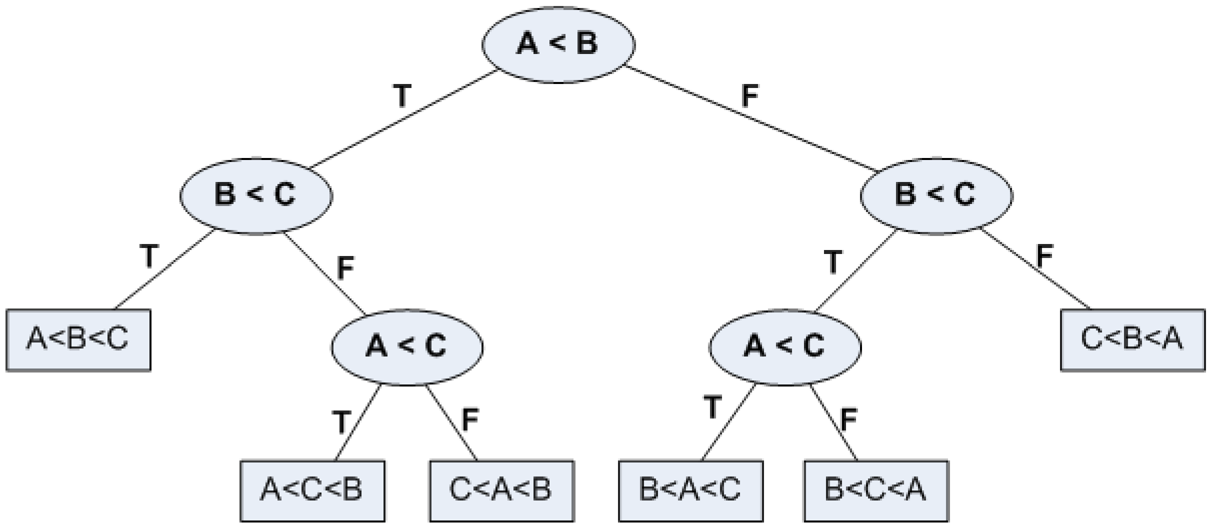
A	*B*	*C*	*f*
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7



Basic Concept

- In Example 18.1, we have considered a decision tree where values of any attribute if binary only. Decision tree is also possible where attributes are of continuous data type

Example 8.2: Decision Tree with numeric data



Some Characteristics

- Decision tree may be *n*-ary, *n* ≥ 2.

- There is a special node called root node.

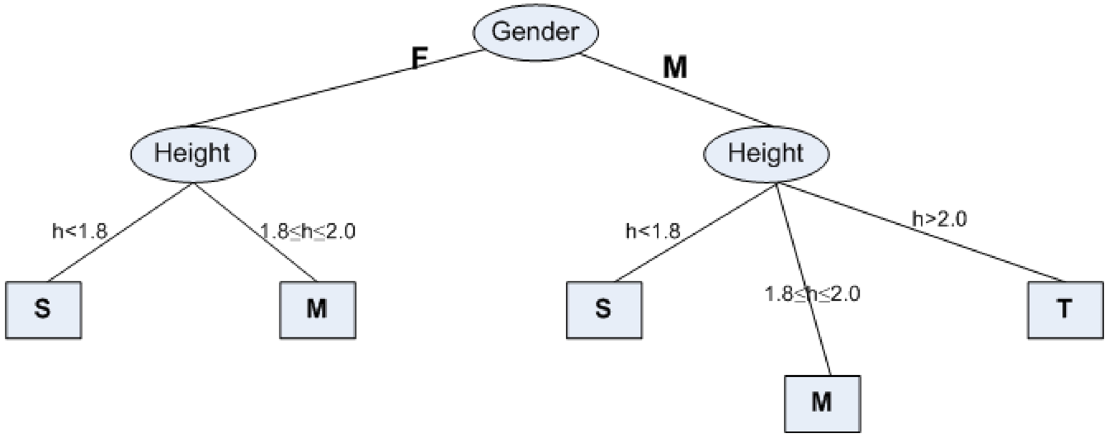
- All nodes drawn with circle (ellipse) are called internal nodes.

- All nodes drawn with rectangle boxes are called terminal nodes or leaf nodes.

- Edges of a node represent the outcome for a value of the node.

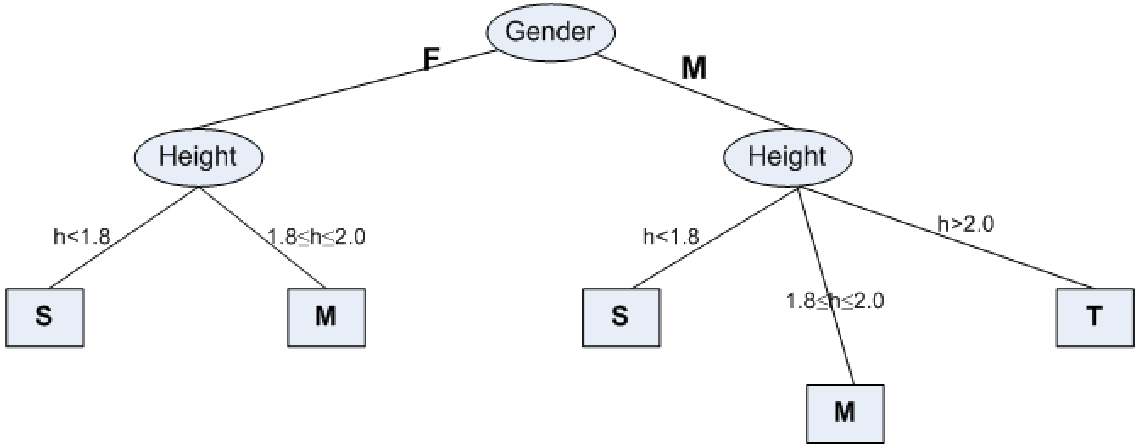
- In a path, a node with same label is never repeated.

- Decision tree is not unique, as different ordering of internal nodes can give different decision tree.



Decision Tree and Classification Task

- Decision tree helps us to classify data.
  - Internal nodes are some attribute
  - Edges are the values of attributes
  - External nodes are the outcome of classification



- Such a classification is, in fact, made by posing questions starting from the root node to each terminal node.

Decision Tree and Classification Task

Example 8.3 : Vertebrate Classification

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class
Human	Warm	hair	yes	no	no	yes	no	**Mammal**
Python	Cold	scales	no	no	no	no	yes	**Reptile**
Salmon	Cold	scales	no	yes	no	no	no	**Fish**
Whale	Warm	hair	yes	yes	no	no	no	**Mammal**
Frog	Cold	none	no	semi	no	yes	yes	**Amphibian**
Komodo	Cold	scales	no	no	no	yes	no	**Reptile**
Bat	Warm	hair	yes	no	yes	yes	yes	**Mammal**
Pigeon	Warm	feathers	no	no	yes	yes	no	**Bird**
Cat	Warm	fur	yes	no	no	yes	no	**Mammal**
Leopard	Cold	scales	yes	yes	no	no	no	**Fish**
Turtle	Cold	scales	no	semi	no	yes	no	**Reptile**
Penguin	Warm	feathers	no	semi	no	yes	no	**Bird**
Porcupine	Warm	quills	yes	no	no	yes	yes	**Mammal**
Eel	Cold	scales	no	yes	no	no	no	**Fish**
Salamander	Cold	none	no	semi	no	yes	yes	**Amphibian**

What are the class label of Dragon and Shark?

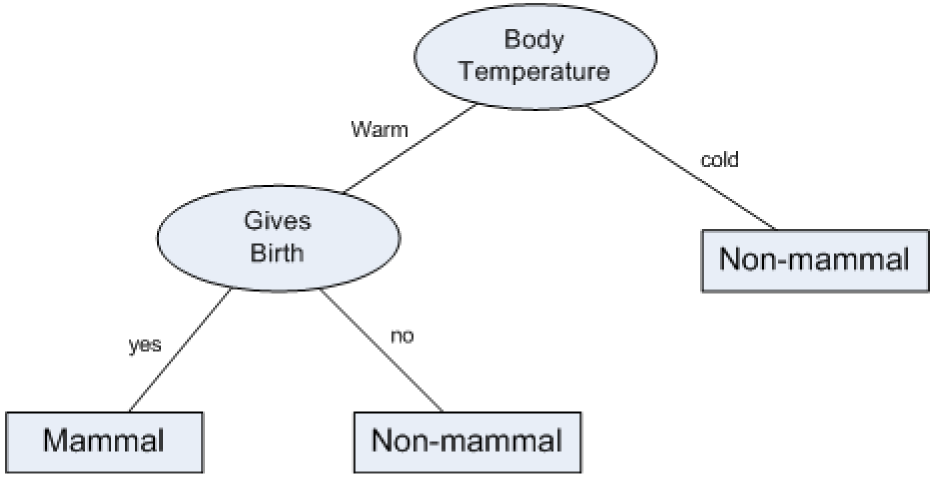
Decision Tree and Classification Task

Example 8.3 : Vertebrate Classification

- Suppose, a new species is discovered as follows.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class
Gila Monster	cold	scale	no	no	no	yes	yes	?

- Decision Tree that can be inducted based on the data (in Example 8.3) is as follows.



Decision Tree and Classification Task

- The above Example illustrates how we can solve a classification problem by asking a series of question about the attributes.
    - Each time we receive an answer, a follow-up question is asked until we reach a conclusion about the class-label of the test.
- The series of questions and their answers can be organized in the form of a decision tree
    - As a hierarchical structure consisting of nodes and edges
- Once a decision tree is built, it is applied to any test to classify it.

Definition of Decision Tree

Definition 1: **Decision Tree**

Given a database *D* = here denotes a tuple, which is defined by a set of attribute set of classes *C* = .

A decision tree ***T*** is a tree associated with *D* that has the following properties:

- Each internal node is labeled with an attribute A_i
- Each edges is labeled with predicate that can be applied to the attribute associated with the parent node of it
- Each leaf node is labeled with class c_j

Building Decision Tree

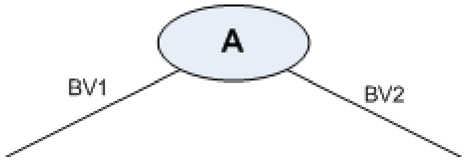
- In principle, there are exponentially many decision tree that can be constructed from a given database (also called training data).
    - Some of the tree may not be optimum
    - Some of them may give inaccurate result

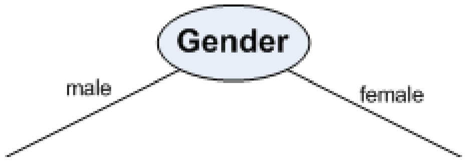
- Two approaches are known
    - **Greedy strategy**
        - A top-down recursive divide-and-conquer
    - **Modification of greedy strategy**
        - ID3
        - C4.5
        - CART, etc.

Node Splitting in **BuildDT** Algorithm

- BuildDT algorithm must provides a method for expressing an attribute test condition and corresponding outcome for different attribute type

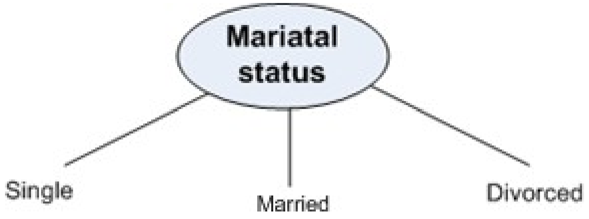
- **Case: Binary attribute**
  - This is the simplest case of node splitting
  - The test condition for a binary attribute generates only two outcomes



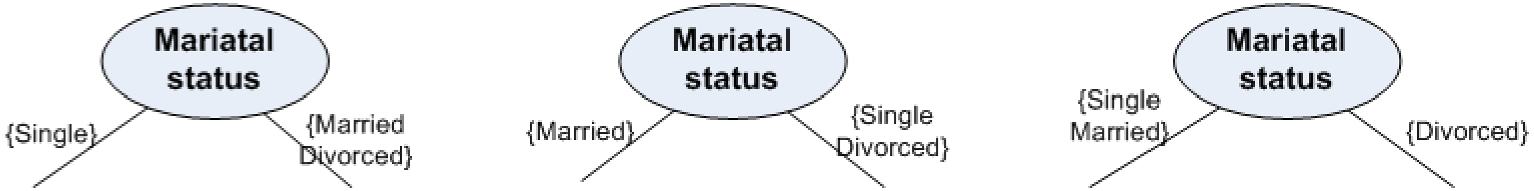


Node Splitting in **BuildDT** Algorithm

- **Case: Nominal attribute**
  - Since a nominal attribute can have many values, its test condition can be expressed in two ways:
    - A multi-way split
    - A binary split
  - Muti-way split: Outcome depends on the number of distinct values for the corresponding attribute



  - Binary splitting by grouping attribute values

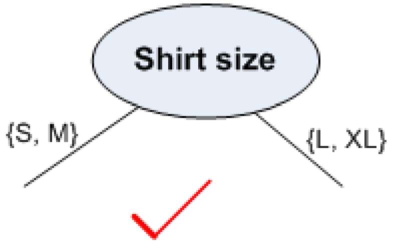


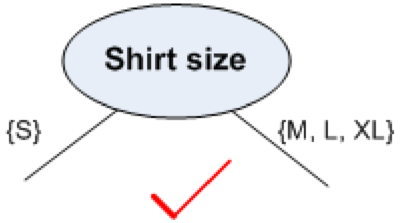
Node Splitting in **BuildDT** Algorithm

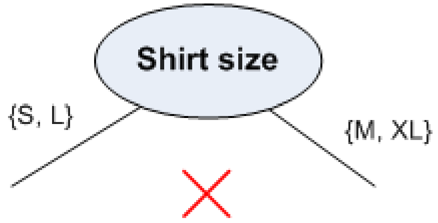
- **Case: Ordinal attribute**
  - It also can be expressed in two ways:
    - A multi-way split
    - A binary split

  - Multi-way split: It is same as in the case of nominal attribute

  - Binary splitting attribute values should be grouped maintaining the order property of the attribute values







Node Splitting in **BuildDT** Algorithm

- **Case: Numerical attribute**
  - For numeric attribute (with discrete or continuous values), a test condition can be expressed as a comparison set
    - Binary outcome: $A > v$ or $A \leq v$
      - In this case, decision tree induction must consider all possible split positions
    - Range query : $v_i \leq A < v_{i+1}$ for $i = 1, 2, \ldots, q$ (if q number of ranges are chosen)
      - Here, q should be decided a priori
  - For a numeric attribute, decision tree induction is a combinatorial optimization problem

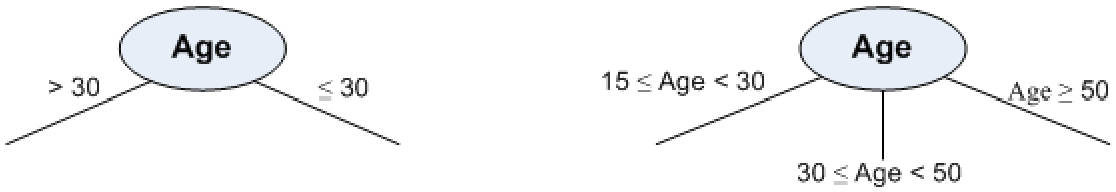


Illustration : **BuildDT** Algorithm

Example 8.4: Illustration of BuildDT Algorithm

- Consider a training data set as shown.

Person	Gender	Height	Class
1	F	1.6	S
2	M	2.0	M
3	F	1.9	M
4	F	1.88	M
5	F	1.7	S
6	M	1.85	M
7	F	1.6	S
8	M	1.7	S
9	M	2.2	T
10	M	2.1	T
11	F	1.8	M
12	M	1.95	M
13	F	1.9	M
14	F	1.8	M
15	F	1.75	S

Attributes:

Gender = {Male(M), Female (F)} // Binary attribute
Height = {1.5, …, 2.5} // Continuous attribute

Class = {Short (S), Medium (M), Tall (T)}

Given a person, we are to test in which class s/he belongs

Illustration : **BuildDT** Algorithm

- To built a decision tree, we can select an attribute in two different orderings: <Gender, Height> or <Height, Gender>

- Further, for each ordering, we can choose different ways of splitting

- Different instances are shown in the following.

- **Approach 1 : <Gender, Height>**

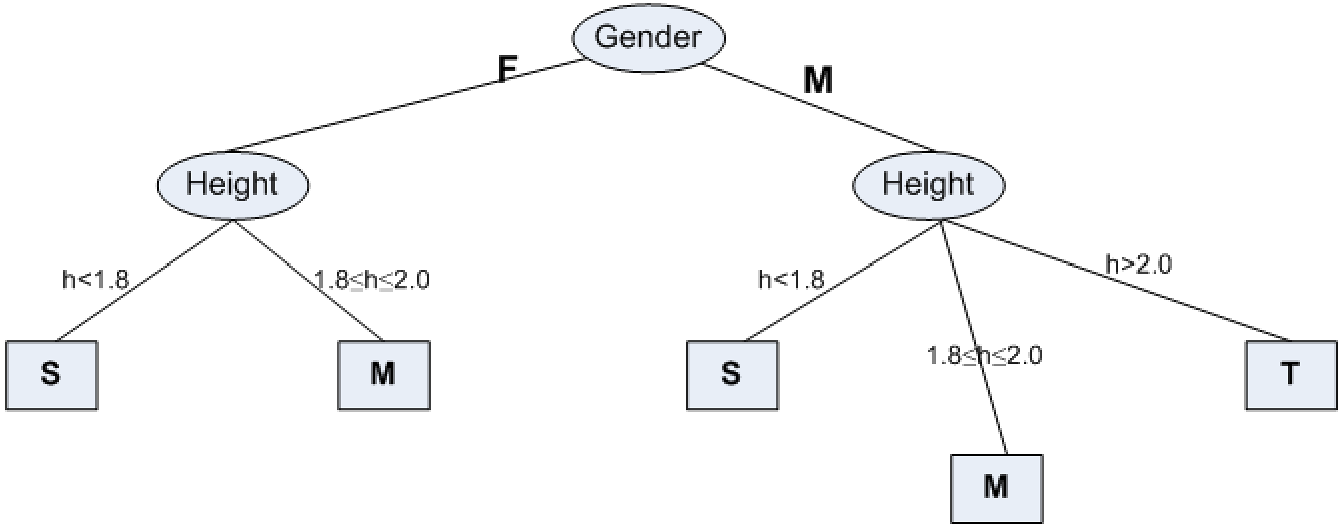
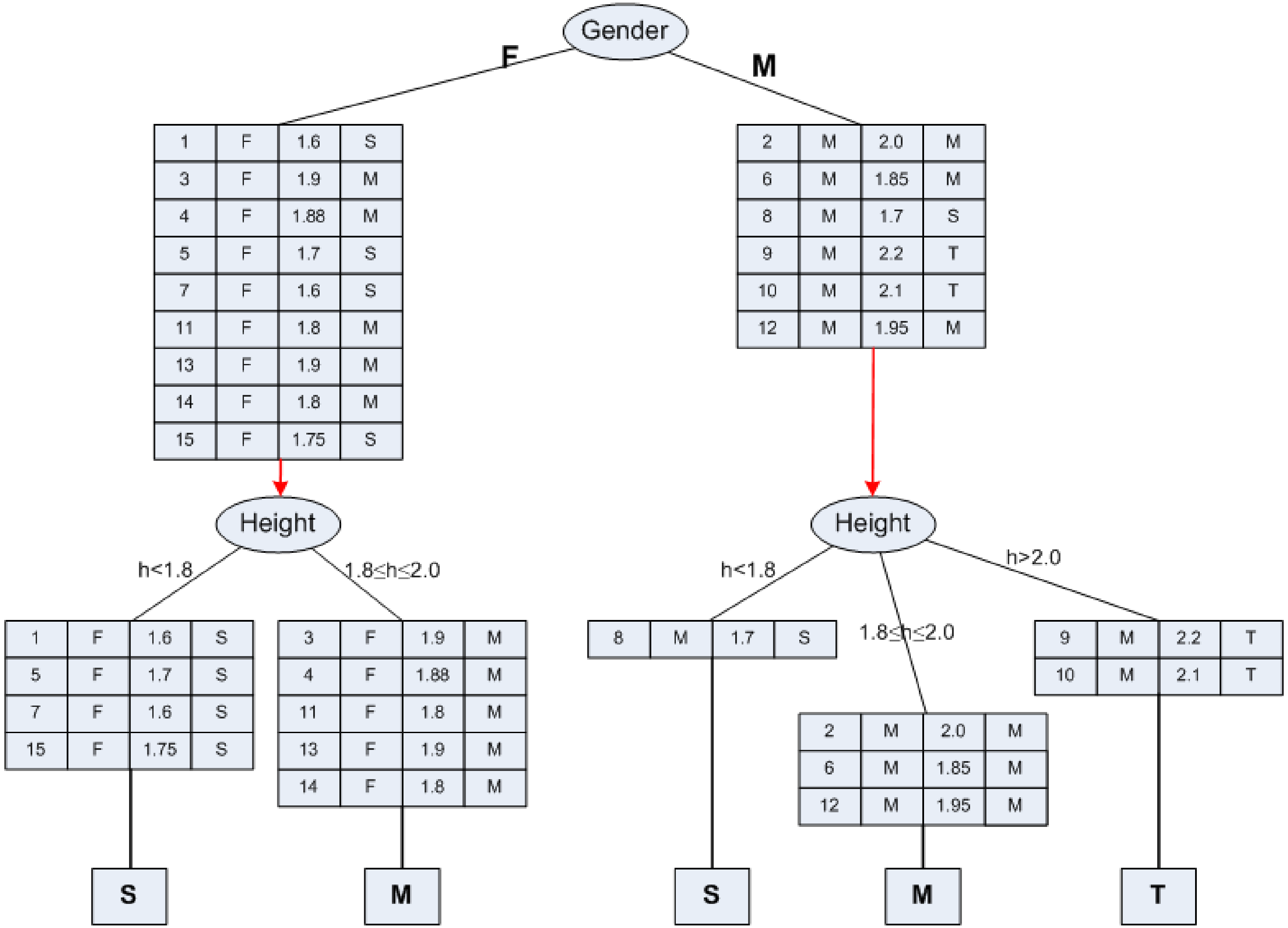


Illustration : **BuildDT** Algorithm



1	F	1.6	S
3	F	1.9	M
4	F	1.88	M
5	F	1.7	S
7	F	1.6	S
11	F	1.8	M
13	F	1.9	M
14	F	1.8	M
15	F	1.75	S

2	M	2.0	M
6	M	1.85	M
8	M	1.7	S
9	M	2.2	T
10	M	2.1	T
12	M	1.95	M

1	F	1.6	S
5	F	1.7	S
7	F	1.6	S
15	F	1.75	S

3	F	1.9	M
4	F	1.88	M
11	F	1.8	M
13	F	1.9	M
14	F	1.8	M

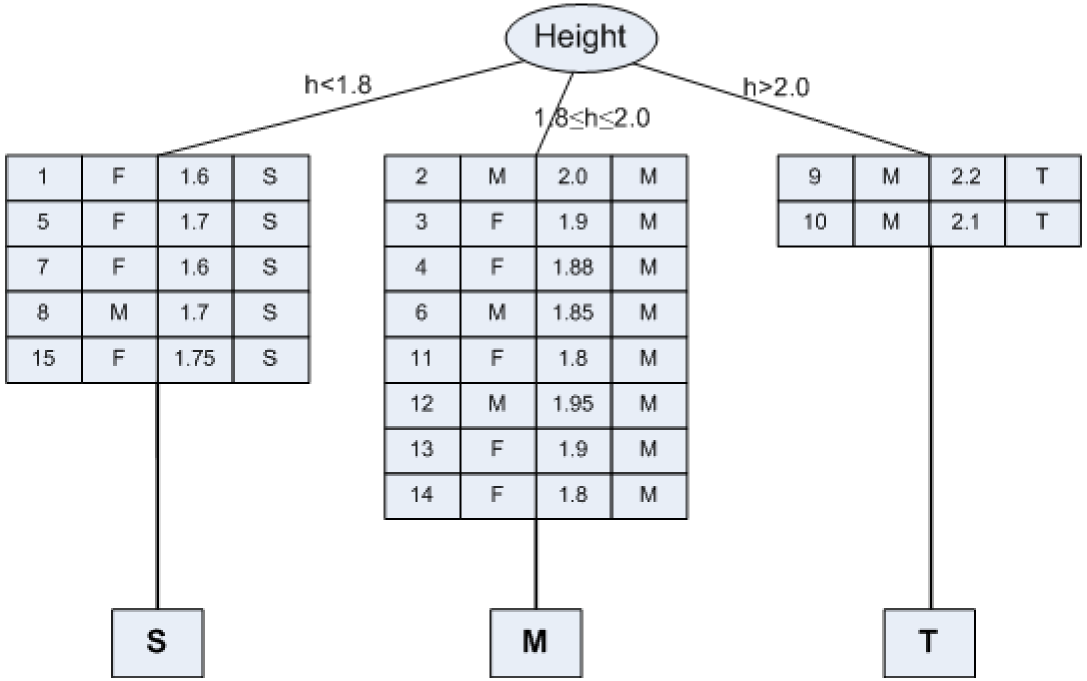
8	M	1.7	S

2	M	2.0	M
6	M	1.85	M
12	M	1.95	M

9	M	2.2	T
10	M	2.1	T

Illustration : **BuildDT** Algorithm

- **Approach 2 : <Height, Gender>**



1	F	1.6	S
5	F	1.7	S
7	F	1.6	S
8	M	1.7	S
15	F	1.75	S

2	M	2.0	M
3	F	1.9	M
4	F	1.88	M
6	M	1.85	M
11	F	1.8	M
12	M	1.95	M
13	F	1.9	M
14	F	1.8	M

9	M	2.2	T
10	M	2.1	T

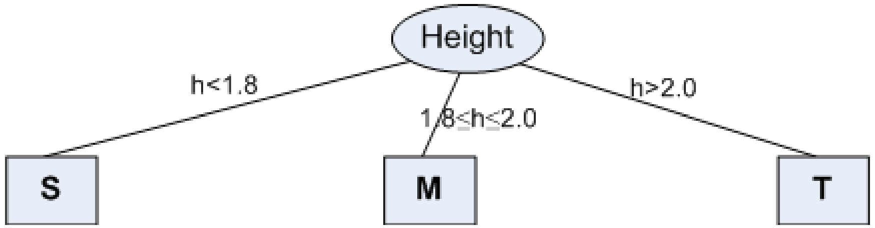


Illustration : **BuildDT** Algorithm

Example 8.5: Illustration of BuildDT Algorithm

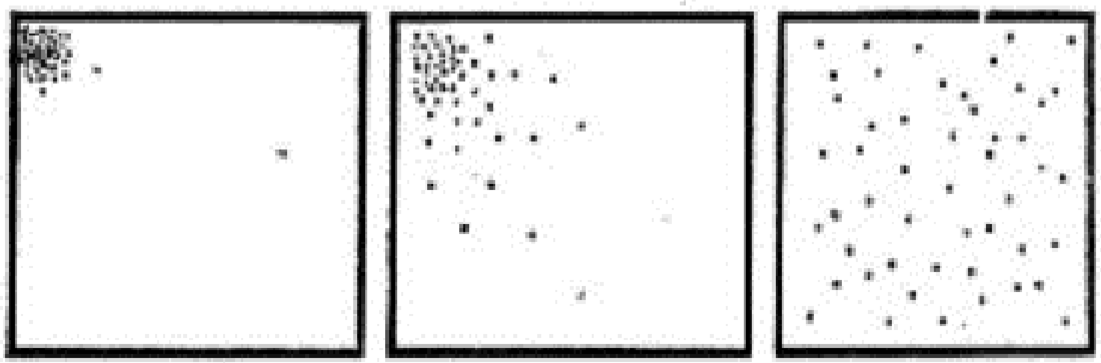
- Consider an anonymous database as shown.

A1	A2	A3	A4	Class
a11	a21	a31	a41	C1
a12	a21	a31	a42	C1
a11	a21	a31	a41	C1
a11	a22	a32	a41	C2
a11	a22	a32	a41	C2
a12	a22	a31	a41	C1
a11	a22	a32	a41	C2
a11	a22	a31	a42	C1
a11	a21	a32	a42	C2
a11	a22	a32	a41	C2
a12	a22	a31	a41	C1
a12	a22	a31	a42	C1

- Is there any "clue" that enables to select the "best" attribute first?
- Suppose, following are two attempts:
  - A1→A2→A3→A4 [naïve]
  - A3→A2→A4→A1 [Random]
- Draw the decision trees in the above-mentioned two cases.
- Are the trees different to classify any test data?
- If any other sample data is added into the database, is that likely to alter the decision tree already obtained?

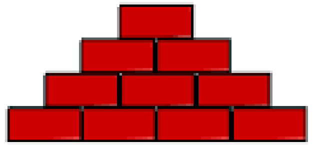
Concept of Entropy

Concept of Entropy

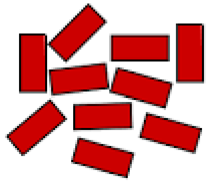


If a point represents a gas molecule, then which system has the more entropy?

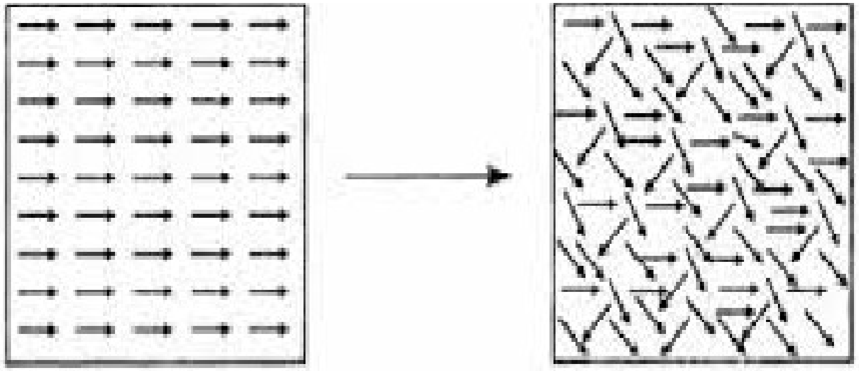
How to measure? ?



More **ordered**
less **entropy**



Less ordered
higher entropy



More organized or
ordered (less **probable**)

Less organized or
disordered (**more** probable)

An Open Challenge!

Roll No.	Assignment	Project	Mid-Sem	End-Sem
12BT3FP06	89	99	56	91
10IM30013	95	98	55	93
12CE31005	98	96	58	97
12EC35015	93	95	54	99
12GG2005	90	91	53	98
12MI33006	91	93	57	97
13AG36001	96	94	58	95
13EE10009	92	96	56	96
13MA20012	88	98	59	96
14CS30017	94	90	60	94
14ME10067	90	92	58	95
14MT10038	99	89	55	93

Roll No.	Assignment	Project	Mid-Sem	End-Sem
12BT3FP06	19	59	16	71
10IM30013	37	38	25	83
12CE31005	38	16	48	97
12EC35015	23	95	54	19
12GG2005	40	71	43	28
12MI33006	61	93	47	97
13AG36001	26	64	48	75
13EE10009	92	46	56	56
13MA20012	88	58	59	66
14CS30017	74	20	60	44
14ME10067	50	42	38	35
14MT10038	29	69	25	33

Two sheets showing the tabulation of marks obtained in a course are shown.

Which tabulation of marks shows the “good” performance of the class?
How you can measure the same?

Entropy and its Meaning

- Entropy is an important concept used in Physics in the context of heat and thereby uncertainty of the states of a matter.

- At a later stage, with the growth of Information Technology, entropy becomes an important concept in Information Theory.

- To deal with the classification job, entropy is an important concept, which is considered as

  - an information-theoretic measure of the “uncertainty” contained in a training data

    - due to the presence of more than one classes.

Entropy in Information Theory

- The entropy concept in information theory first time coined by Claude Shannon (1850).

- The first time it was used to measure the “information content” in messages.

- According to his concept of entropy, presently entropy is widely being used as a way of representing messages for efficient transmission by Telecommunication Systems.

Measure of Information Content

- People, in general, are information hungry!
- Everybody wants to acquire information (from newspaper, library, nature, fellows, etc.)
  - Think how a crime detector do it to know about the crime from crime spot and criminal(s).
  - Kids annoyed their parents asking questions.
- In fact, fundamental thing is that we gather information asking questions (and decision tree induction is no exception).
- We may note that information gathering may be with certainty or uncertainty.

Measure of Information Content

Example 8.6

a) Guessing a birthday of your classmate

It is with uncertainty ~

Whereas guessing the day of his/her birthday is .

This uncertainty, we may say varies between 0 to 1, both inclusive.

b) As another example, a question related to event with eventuality (or impossibility) will be answered with 0 or 1 uncertainty.

- Does sun rises in the East? (answer is with 0 uncertainty)
- Will mother give birth to male baby? (answer is with ½ uncertainty)
- Is there a planet like earth in the galaxy? (answer is with an extreme uncertainty)

Entropy Calculation

- If there are *m* objects with frequencies , ……., , then the average number of bits (i.e. questions) that need to be examined a value, that is, entropy is the frequency of occurrence of the value multiplied by the number of bits that need to be determined, summed up values of from *1* to *m*.

Theorem: Entropy calculation

If p_i denotes the frequencies of occurrences of *m* distinct objects, then the entropy *E* is

Note:

- If all are equally likely, then and ; it is the special case.

Entropy of a Training Set

- If there are *k* classes , ……., and for denotes the number of occurrences of classes divided by the total number of instances (i.e., the frequency of occurrence of) in the training set, then entropy of the training set is denoted by

Here, *E* is measured in “bits” of information.

Note:

- The above formula should be summed over the non-empty classes only, that is, classes for which
- *E* is always a positive quantity
- *E* takes it’s minimum value (zero) if and only if all the instances have the same class (i.e., the training set with only **one** non-empty class, for which the probability 1).
- Entropy takes its maximum value when the instances are equally distributed among *k* possible classes. In this case, the maximum value of *E* is .

Entropy of a Training Set

Example 18.10: OPTH dataset

Consider the OTPH data shown in the following table with total 24 instances in it.

Age	Eye sight	Astigmatic	Use Type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

A coded forms for all values of attributes are used to avoid the cluttering in the table.

Entropy of a training set

Specification of the attributes are as follows.

Age	Eye Sight	Astigmatic	Use Type
1: Young	1: Myopia	1: No	1: Frequent
2: Middle-aged	2: Hypermetropia	2: Yes	2: Less
3: Old			

Class: 1: Contact Lens 2:Normal glass 3: Nothing

In the OPTH database, there are 3 classes and 4 instances with class 1, 5 instances with class 2 and 15 instances with class 3. Hence, entropy *E* of the database is:

Note:

- The entropy of a training set implies the number of yes/no questions, on the average, needed to determine an unknown test to be classified.
- It is very crucial to decide the series of questions about the value of a set of attribute, which collectively determine the classification. Sometimes it may take one question, sometimes many more.
- Decision tree induction helps us to ask such a series of questions. In other words, we can utilize entropy concept to build a better decision tree.

How entropy can be used to build a decision tree ?

Decision Tree Induction Techniques

- Decision tree induction is a top-down, recursive and divide-and-conquer approach.
- The procedure is to choose an attribute and split it into from a larger training set into smaller training sets.

- Different algorithms have been proposed to take a good control over
  1. Choosing the best attribute to be splitted, and
  2. Splitting criteria

- Several algorithms have been proposed for the above tasks. In this lecture, we shall limit our discussions into three important of them
  - **ID3**
  - **C 4.5**
  - **CART**

Algorithm ID3

ID3: Decision Tree Induction Algorithms

- Quinlan [1984] introduced the ID3, a popular short form of **I**terative **D**ichotomizer 3 for decision trees from a set of training data.

- In ID3, each node corresponds to a splitting attribute and each arc is a possible value of that attribute.

- At each node, the splitting attribute is selected to be the most informative among the attributes not yet considered in the path starting from the root.

Algorithm ID3

- In ID3, entropy is used to measure how informative a node is.
    - It is observed that splitting on any attribute has the property that average entropy of the resulting training subsets will be less than or equal to that of the previous training set.
- ID3 algorithm defines a measurement of a splitting called **Information Gain** to determine the goodness of a split.
    - The attribute with the largest value of information gain is chosen as the splitting attribute and
    - it partitions into a number of smaller training sets based on the distinct values of attribute under split.

Defining Information Gain

- We consider the following symbols and terminologies to define information gain, which is denoted as α.

- *D* denotes the training set at any instant

- |*D*| denotes the size of the training set *D*

- *E*(*D*) denotes the entropy of the training set *D*

- The entropy of the training set *D*

$$E(D) = -$$

  - where the training set *D* has , , … , , the *k* number of distinct classes and
  - , 0< is the probability that an arbitrary tuple in *D* belongs to class (*i* = *1, 2, … , k).*

Defining Information Gain

Definition 2: **Weighted Entropy**

The weighted entropy denoted as $E_A(D)$ for all partitions of D with respect to A is given by:

$= E()$

Here, the term denotes the weight of the j-th training set.

More meaningfully, $E_A(D)$ is the expected information required to classify a tuple from D based on the splitting of A.

Defining Information Gain

- Our objective is to take *A* on splitting to produce an exact classification (also called pure), that is, all tuples belong to one class.

- However, it is quite likely that the partitions is impure, that is, they contain tuples from two or more classes.

- In that sense, is a measure of impurities (or purity). A lesser value of implying more power the partitions are.

Definition 3: **Information Gain**

Information gain, of the training set *D* splitting on the attribute *A* is given by

$= E(D) -$

In other words, gives us an estimation how much would be gained by splitting on *A*. The attribute *A* with the highest value of should be chosen as the splitting attribute for *D*.

Information Gain Calculation

Example 8.11 : Information gain on splitting OPTH

- Let us refer to the OPTH database discussed earlier.

- Splitting on **Age** at the root level, it would give three subsets and as shown in the tables in the following three slides.

- The entropy and of training sets and and corresponding weighted entropy and are also shown alongside.

- The Information gain is then can be calculated as **0.0394**.

- Recall that entropy of OPTH data set, we have calculated as *E(OPTH)* = **1.3261**

  (see Slide #16)

Information Gain Calculation

Example 8.11 : Information gain on splitting OPTH

Training set: (Age = 1)

Age	Eye-sight	Astigmatism	Use type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1

() ()

() = **1.5**

× 1.5 = **0.5000**

Calculating Information Gain

Training set: (Age = 2)

Age	Eye-sight	Astigmatism	Use type	Class
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3

$() \, () \;\; ()$

$= \mathbf{1.2988}$

$\times\ 1.2988 = \mathbf{0.4329}$

Calculating Information Gain

Training set: (Age = 3)

Age	Eye-sight	Astigmatism	Use type	Class
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

() ()

$() \mathbf{= 1.0613}$

$\times 1.0613 \mathbf{= 0.3504}$

$= 1.3261 – (0.5000 + 0.4329 + 0.3504) = \mathbf{0.0394}$

Information Gains for Different Attributes

- In the same way, we can calculate the information gains, when splitting the OPTH database on Eye-sight, Astigmatic and Use Type. The results are summarized below.
- Splitting attribute: Age
- Splitting attribute: Eye-sight
- Splitting attribute: Astigmatic

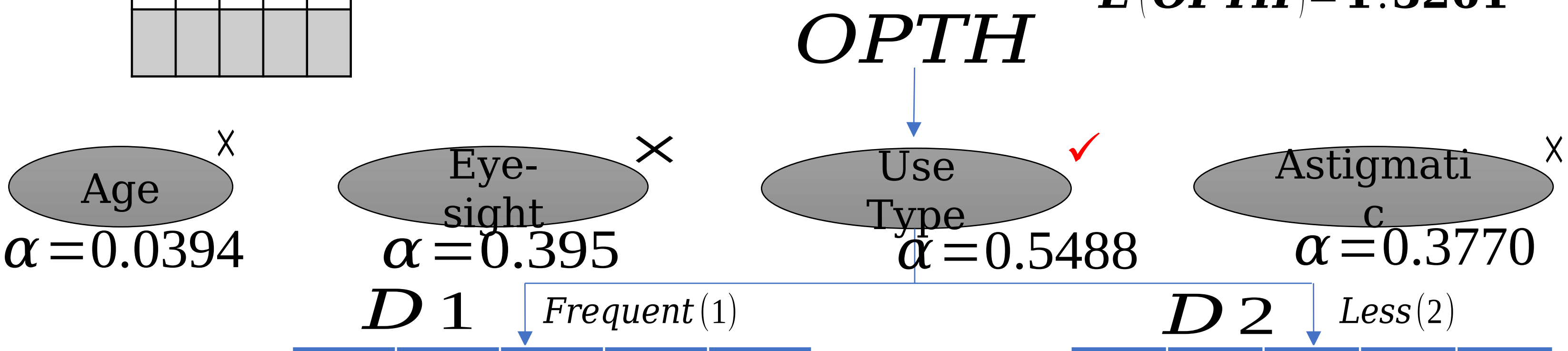
  770
- Splitting attribute: Use Type

  5488

Decision Tree Induction : ID3 Way

- The ID3 strategy of attribute selection is to choose to split on the attribute that gives the greatest reduction in the weighted average entropy
  - The one that maximizes the value of information gain
- In the example with OPTH database, the larger values of information gain is 5488
  - Hence, the attribute should be chosen for splitting is “Use Type”.
- The process of splitting on nodes is repeated for each branch of the evolving decision tree, and the final tree, which would look like is shown in the following slide and calculation is left for practice.

Decision Tree Induction : ID3 Way



$E(OPTH)=1.3261$

$OPTH$

Age ✗ $\alpha=0.0394$

Eye-sight ✗ $\alpha=0.395$

Use Type ✓ $\alpha=0.5488$

Astigmatic ✗ $\alpha=0.3770$

$D1$ $Frequent(1)$

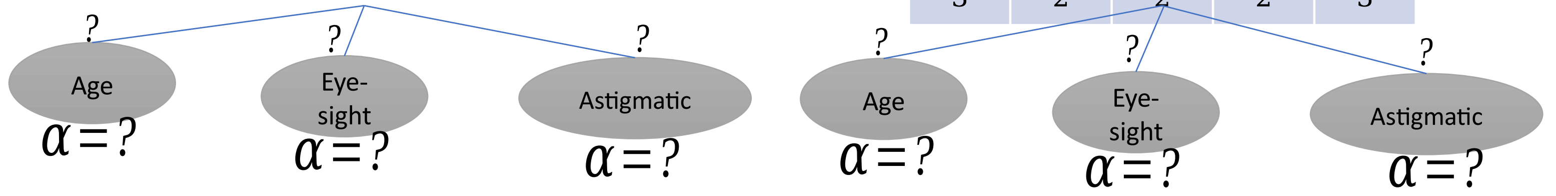
$D2$ $Less(2)$

E(D¿¿1)=?¿

Age	Eye	Ast	Use	Class
1	1	1	1	3
1	1	2	1	3
1	2	1	1	3
1	2	2	1	3
2	1	1	1	3
2	2	1	1	3
2	2	2	1	3
3	1	1	1	3
3	1	2	1	3
3	2	1	1	3
3	2	2	1	3

Age	Eye	Ast	Use	Class
1	1	1	2	2
1	1	2	2	1
1	2	1	2	2
1	2	2	2	1
2	1	1	2	2
2	1	2	2	1
2	2	1	2	2
3	1	1	2	3
3	1	2	2	3
3	2	1	2	2
3	2	2	2	3

E(D¿¿2)=?¿



Age ? $\alpha=?$

Eye-sight ? $\alpha=?$

Astigmatic ? $\alpha=?$

Age ? $\alpha=?$

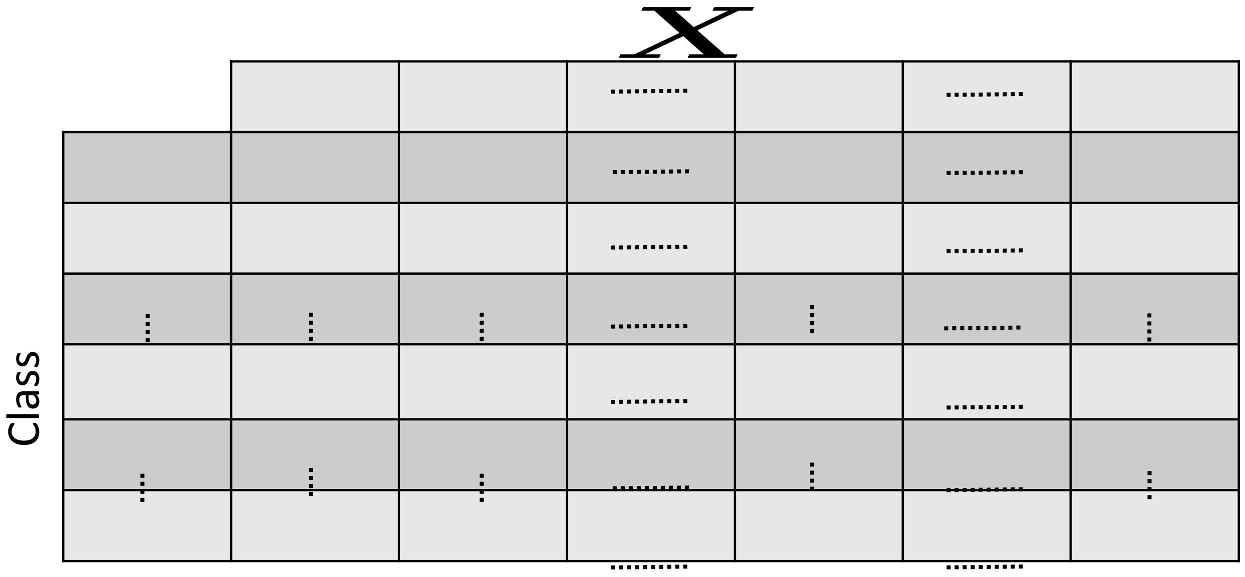
Eye-sight ? $\alpha=?$

Astigmatic ? $\alpha=?$

Frequency Table : Calculating α

- Calculation of entropy for each table and hence information gain for a particular split appears tedious (at least manually)!
- As an alternative, we discuss **a short-cut method** of doing the same using a special data structure called **Frequency Table.**
- **Frequency Table**: Suppose, denotes an attribute with attribute values in it. For a given database , there are a set of say . Given this, a frequency table will look like as follows.

Frequency Table : Calculating α



- Number of rows = Number of classes
- Number of columns = Number of attribute values
- = Frequency of for class

Assume that , the number of total instances of .

Calculation of α using Frequency Table

Example 8.12 : OTPH Dataset

With reference to OPTH dataset, and for the attribute Age, the frequency table would look like

	Age=1	Age=2	Age=3	Row Sum
Class 1	2	1	1	4
Class 2	2	2	1	5
Class 3	4	5	6	15
Column Sum	8	8	8	24

N=24

Calculation of α using Frequency Table

- The weighted average entropy then can be calculated from the frequency table following the
  - Calculate for all *(Entry Sum)* and
  - Calculate for all *(Column Sum)* in the row of column sum
  - Calculate

Example 8.13: OTPH Dataset

For the frequency table in **Example 18.12**, we have

Limiting Values of Information Gain

- The Information gain metric used in ID3 always should be positive or zero.

- It is always positive value because information is always gained (i.e., purity is improved) by splitting on an attribute.

- On the other hand, when a training set is such that if there are classes, and the entropy of training set takes the largest value i.e., (this occurs when the classes are balanced), then the information gain will be zero.

Limiting Values of Information Gain

Example 8.14: Limiting values of Information gain

Consider a training set shown below.

Data set *Table A*

X	Y	Class
1	1	A
1	2	B
2	1	A
2	2	B
3	2	A
3	1	B
4	2	A
4	1	B

X *Table X*

	1	2	3	4
A	1	1	1	1
B	1	1	1	1
	2	2	2	2

Frequency table of X

Y *Table Y*

	1	2
A	2	2
B	2	2
	4	4

Frequency table of Y

Limiting values of Information Gain

- Entropy of Table A is

(The maximum entropy).

- In this example, whichever attribute is chosen for splitting, each of the branches will also be balanced thus each with maximum entropy.
- In other words, information gain in both cases (i.e., splitting on X as well as Y) will be zero.

Note:

- The absence of information gain does not imply that there is no profit for splitting on the attribute.
- Even if it is chosen for splitting, ultimately it will lead to a final decision tree with the branches terminated by a leaf node and thus having an entropy of zero.
- Information gain can never be a negative value.

Reference

- The detail material related to this lecture can be found in

Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014

Any question?